Hindawi
Wireless Communications and Mobile Computing
Volume 2022, Article ID 6177973, 10 pages
https://doi.org/10.1155/2022/6177973

Research Article

Online Multiplayer Tracking by Extracting Temporal Contexts with Transformer

Xiao Han,[1,2] **Yongbin Wang**,[2,3] **Shouxun Liu**,[1] **and Cong Jin**[1]

[1]*School of Information and Communication Engineering, Communication University of China, Beijing, China*
[2]*State Key Laboratory of Media Convergence and Communication, Communication University of China, Beijing, China*
[3]*Collaborative Innovation Center, Communication University of China, Beijing, China*

Correspondence should be addressed to Yongbin Wang; ybwang@cuc.edu.cn

Received 31 October 2021; Revised 4 August 2022; Accepted 13 September 2022; Published 11 October 2022

Academic Editor: Ting Bi

Sports competition is one of the most popular programs for many audiences. Tracking the players in sports game videos from broadcasts is a nontrivial challenge for computer vision researchers. In sports videos, the direction of an athlete's movement changes quickly and unpredictably. Mutual occlusion between athletes is also more frequent in team competitions. However, the rich temporal contexts among the adjacent frames have been excluded from consideration. To address this dilemma, we propose an online transformer-based learnable framework in an end-to-end fashion. We use a transformer architecture to extract the temporal contexts between the successive frames and add them to the network training, which is robust to occlusion and complex direction changes in multiplayer tracking. We demonstrate the effectiveness of our method on three sports video datasets by comparing them with recently advanced multiplayer trackers.

1. Introduction

With the increasing number of spectators who prefer sports games, video analysis of sports games has received more and more increasing attention in the field of computer vision. Multiple player tracking has become an urgent demand for sports video analysis, which has substantial benefits for intelligently editing game videos. For example, the identity information and movement information of athletes obtained by automatic tracking can be visualized and quantified. The audience can quickly focus on interested athletes and obtain more interactivity from the sports game. If tracking information can be obtained in real time, broadcasters can use them to assist in broadcasting, editing the highlight clips, and commenting on the game. The scope and amount of movement for each player can provide objective clues for assessing the player's abilities and formulating specific strategies. It is essential to correctly track multiple players even under various challenging conditions.

Several efforts have been made to address this issue. The traditional method uses Bayesian inference to solve the association problem. For instance, associating the identities of the isolated tracks by exploiting the graph constraints and similarity measures [1]. They formulate it as a Bayesian network inference problem. Reference [2] proposed a dual-mode two-way Bayesian inference approach that dynamically switches between an offline general model and an online dedicated model to address single isolated object tracking and multiple occluded objects tracking integrally by forward filtering and backwards smoothing. With the development of CNNs, researchers have begun to use deep learning to extract features. Reference [3] first utilized Faster R-CNN [4] to generate an initial detection, and the associating step is modeled as a minimum-cost network flow problem. An adaptive multiple scale sampling scheme based on spatially proximate foreground regions [5] is very helpful for preserving the underlying states of tracked objects even with severe occlusions. Despite some success, multiplayer tracking has a few problems. For example, most of the previous methods are offline, which cannot meet real-time requirements due to the use of information after the current frame. In addition, they all follow the standards of tracking by detection. Having the two models separately executed may lead to efficiency problems. In

the real scene, there are still some difficulties in sports game video broadcasts. In the case of live broadcasts, athletes need to be tracked in real time so that player analysis and highlight editing can be quickly presented to the audience.

Multiobject tracking generally refers to pedestrians or vehicles. Compared with multiobject tracking where the movement direction is relatively stable, multiplayer tracking has additional difficulties. Multiplayer tracking has rapid and frequent changes in the direction of a player's movement, and there are frequent occlusion and disappearance-reproduction problems in team sports. To effectively deal with these problems, most of the previous multiplayer tracking methods are offline and not end-to-end. In this paper, we propose an end-to-end online multiplayer tracking model using a transformer structure to extract the temporal domain information between adjacent frames and add it to the model training, which better solves the occlusion and the problem of sudden changes in direction.

Our contribution in this work can be summarized as follows:

(a) *Temporal contextual information among the successive frames is optimized by the transformer structure*

(b) *The previous frames' feature map is reused as an input of the current frame encoder, which can quickly and effectively associate the tracking box and reduce the missed tracking*

(c) *Our method is online and end-to-end, which is more concise and robust compared with previous multiplayer tracking models*

We use the MOT challenge evaluation metrics to perform comparative experiments on three sports datasets. Our model performs well on some indicators.

2. Related Work

2.1. Multiple Object Tracking. In the generalized MOT, the tracked objects include pedestrians, vehicles, animals, athletes, cells, and some rigid objects. MPT is a more specific task of research objects in the field of MOT. We first review the related work on player tracking, including both multiobject tracking and multiplayer tracking. Then, we briefly discuss the differences between multiple object tracking (MOT) and multiple player tracking (MPT). Finally, we also discuss the application of transformer architecture in the computer vision community.

Object tracking is a vital and basic task in computer vision. It has been applied in various real-world areas, such as security monitoring systems, autonomous driving, and video understanding. Single-object tracking (SOT) mainly provides the position information of the object to be tracked in the first frame and locates the object in the following consecutive video frames. However, MOT has no prior knowledge. The position coordinates of all objects in each frame and the corresponding identity ID to each object need to be labeled in the MOT task to distinguish the objects in the inner classification.

The current mainstream MOT algorithms are divided into two categories. The tracking-by-detection method, such as SORT [6] and Deep SORT [7]. First, a series of bounding boxes are extracted through conventional object detection methods, and then, based on the relationship between the previous and subsequent frames, the bounding boxes containing the same object are assigned the same ID. Recently, there have also been many studies on jointly learning the detector and data association, including JDE [8] and FairMOT [9]. Our transformer-based method allows object detection and appearance embedding to be learned in an end-to-end model. Thanks to the attention architecture in the transformers [10], the method we propose can learn efficiently and obtain the bounding box and identity ID simultaneously.

The MOT algorithm can also be divided into offline and online methods. When trying to determine the object location and ID information in a certain frame, the offline tracking algorithm [11–13] can use the information after the current frame. Because of the availability of more global information, offline algorithm results are often more accurate, but they generally consume more time. In contrast, online tracking algorithms [14–17] can only use current and past information to predict the current frame. The online tracking method is very suitable for automatic driving, navigation, program live broadcasting, and other tasks that require high real-time performance. Compared with offline methods, the performance of online methods tends to be less accurate because the methods cannot use future information to repair previous errors. To some extent, the multiplayer tracking studied in this paper belongs to MOT. However, compared with ordinary pedestrian and vehicle tracking, multiple player tracking has difficulties, such as a more similar appearance within the class, unstable movement direction and speed, and more occlusions and collisions. By making use of the attention's unique structure, which correlates the entire input sequence, we carefully modify the classic transformer structure to obtain an online multiplayer tracking model. Additionally, our model is in an end-to-end form.

2.2. Multiple Player Tracking. Some contributions have been proposed in previous studies on soccer player tracking [18, 19]. However, as mentioned above, there are three main differences between player tracking and pedestrian tracking.

(1) The appearance of the athletes is more similar. Most pedestrian clothes are diverse in color and style. In contrast, athletes, especially those from the same team, wear the same uniform. However, the jersey number can be used as a clue feature to distinguish the players

(2) Pedestrians generally move in a uniform speed and in a straight line, while the athlete's movement direction and speed are unpredictable due to their frequent drastic swerves and sudden speed changes, which will increase the difficulty of tracking due to severe deformation

(3) In addition, compared to pedestrians, athletes will crash into each other and collide more frequently

To resolve these difficult problems, previous researchers have made some developments. Pallavi et al. [18] is a tracking-by-detection method that solves the problems of player detection, labeling, and tracking in broadcast soccer videos. To accommodate the frequent disappearance and reappearance of targets, [20] views the date association in multiplayer tracking as a Markov Chain Monte Carlo (MCMC) problem. A dual-mode two-way Bayesian inference approach was proposed in [2] that dynamically switches between an offline general model and an online dedicated model to deal with single isolated object tracking and multiple occluded players tracking integrally by forward filtering and backwards smoothing. Liu et al. [21] extracted a set of "Game Context Features" from the video with off-field interference removed to represent the current state of the field. Then, a random decision forest combining the current trajectory and the context features selects the best affinity model for a certain athlete at a certain moment.

2.3. Transformer. The transformer structure was first proposed in the machine translation task. Transformer architectures are based on a self-attention mechanism that learns the relationships between elements of a sequence. In [10], the transformer abandons the traditional CNN and RNN, and the entire network uses the attention mechanism. More precisely, the transformer only consists of self-attention and a feedforward neural network. In contrast from the sequential structure of the RNN, the parallel computing system of the transformer structure has better parallelism and conforms to the GPU framework. Since then, the transformer model has gained increasing popularity in NLP tasks, such as text classification, machine translation, and question answering. Breakthroughs from transformer networks in Natural Language Processing (NLP) domain have sparked great interest in the computer vision community to adapt these models for vision tasks. Transformers are gradually being used in many vision tasks, such as image recognition, image enhancement, target detection, and image segmentation. To bridge the gap between the fields, many studies have made some modifications when introducing the transformers and their variant transfer learning into visual tasks. For example, [22] focused on completely migrating the transformer to the image classification task and completely abandoned the CNN. The input image is divided into patches, and then, each patch is flattened. The subsequent operation is similar to the BERT [23] in machine translation. Based on the CNN and the transformer, [24] completely removed the postprocessing steps of the previous detection algorithms that rely on artificial a priori for NMS, anchor generators, and constructs a completely end-to-end target detection framework. The aforementioned methods merely use the transformer structure in image-level vision tasks. After that, [25] also introduced transformers to the visual tracking community for the first time. By virtue of the key-query mechanism in the attention architecture, they can track new targets in the joint detection and tracking pipeline. Inspired by [25], we carefully adjusted the structure of the transformer to adapt to the task of multiplayer tracking, and we obtained a competitive result.

3. Approach

As mentioned above, MPT can be regarded as a MOT problem where the tracking objects are athletes, and the purpose is to obtain the position coordinates and identity information of all athletes in consecutive frames. We offer a mathematical formulation of MPT. Given an image frames sequence as the input, devoted by $I = \{I_1, I_2, \cdots, I_t, \cdots\}$, where I_t is the tth frame. We employ $S_t = \{S_t^1, S_t^2, \cdots, S_t^i, \cdots, S_t^m\}$ to devote the state of the tth frame, where m represents the total number of athletes in the tth frame, and S_t^m is the state of the ith athlete in the tth frame. Athlete's state include position, size, speed, direction, and appearance. The trained model is given a sequence of frames and outputs the trajectory T, identity d, and position and size information (x, y, w, h) of all athlete targets in each frame. We employ $p_t^i = (T, d, x, y, w, h)$ to denote the output result of the ith athlete in the tth frame.

To solve the serious deformation and similar appearance of athletes in the MPT task, it is necessary to obtain more accurate identity information and position coordinates of the athletes. In this section, we introduce our proposed model in detail. After that, the settings in the training and inference process are discussed.

3.1. Framework. There are four core components in our model, which are the backbone network to extract the feature map, the encoder component, the decoder tracking component, and the matching component, as shown in Figure 1.

Thanks to the transformer architecture [10], we can exploit the rich temporal contexts among the adjacent frames in the video flow via an encoder-decoder structure. An overview of our transformer-based architecture is illustrated in Figure 2. The encoder part is simple and the same as the classic transformer encoder structure. The encoder takes the feature maps of two consecutive frames as a pair of inputs. The feature map calculated in the previous frame is retained and reused as part of the input of the current encoder, which can reduce computational consumption. Specifically, in the self-attention structures, we use the classic dot-product to calculate the correlation. The add and norm represent residual connection and instance normalization [26], respectively. In the decoder module, we use the same decoder structure to complete object detection and tracking propagation. The difference between the two decoder operations is the input query. The detection branch is the same as DETR [24], which is the learned query, while the query of the tracking propagation branch is the feature of the target provided by the tracker in the previous frame. In the decoding process, the cross-attention block bridges features from the previous frame and the current frame to propagate temporal contexts.

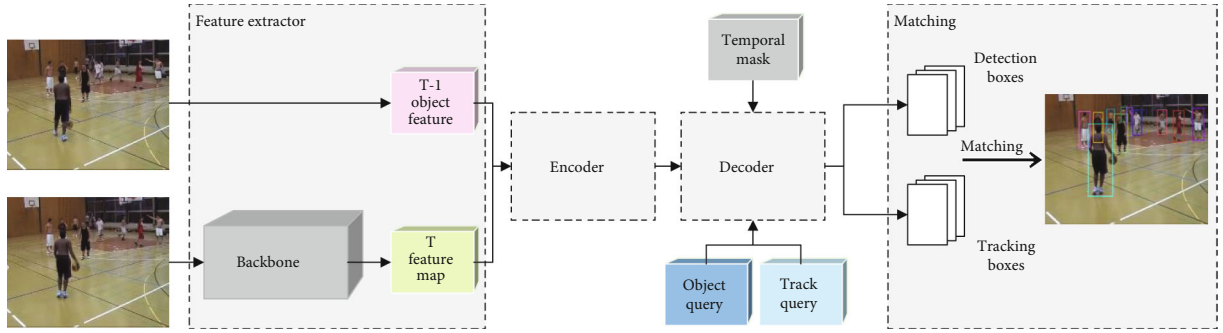

Figure 1: The transformer-based framework of the proposed. The CNN module is used to extract the features of the input frame. The global feature maps of the previous frame and the current frame are fed to the encoder, and then, the combined global feature map of two consecutive frames is input into the decoder as a common key. The temporal mask is beneficial for suppressing the background changes transformed from the previous frame temporally and concentrates on the target player. The object detection features of the current frame and the tracked object features of the previous frame are input into two decoders with a shared structure, and then, we obtain detection boxes and tracking boxes. Finally, IoU matching is employed to associate detection boxes with tracking boxes.

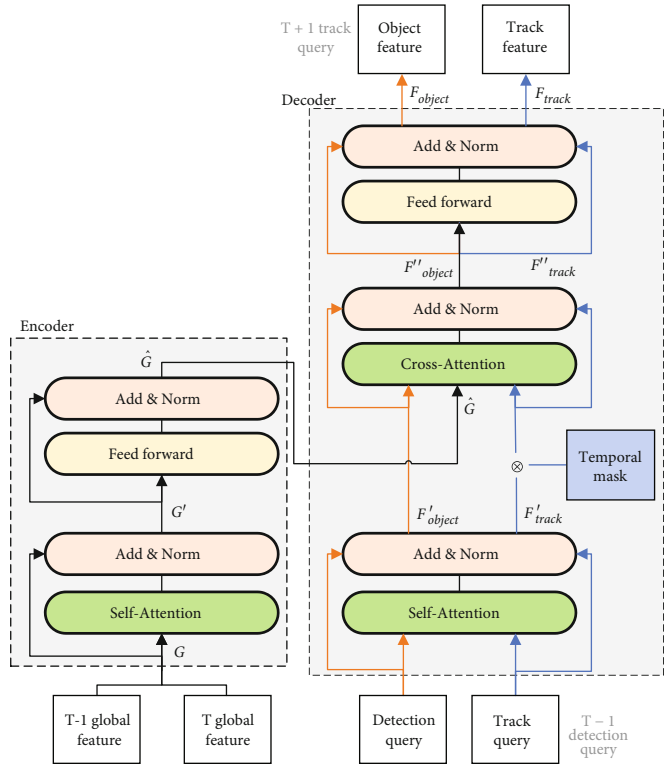

Figure 2: Transformer architecture in our method.

(1) *Encoder*

In the encoder part, the global feature map g_{t-1} of the previous frame is retained. We combine g_{t-1} and the global feature map of the current frame g_t in the series, denoted by G. The similarity matrix $A_{G\longrightarrow G}$ is calculated by the self-attention block as follows:

$$A_{G\longrightarrow G} = \text{Attention}(G, G). \tag{1}$$

Attention(G, G) represents the self-attention operation, as shown in the green box in Figure 2. Then, the assembled global feature G is transformed through $A_{G\longrightarrow G}G$. As shown in Figure 2, the next operation is residual connection and instance normalization as follows:

$$G' = \text{Ins.Norm}(A_{G\longrightarrow G}G + G). \tag{2}$$

We use instance normalization, denoted by Ins.Norm in our encoder-decoder structure. In the following experiments, we compare several mainstream normalization methods. Experimental results show that instance normalization is the best performer among them. G' is the output of the hidden layer after residual connection and instance

normalization. According to [24], the encoder and decoder include a fully connected feed forward network after the attention block. The feed forward network consists of two linear transformations with a ReLU activation in between as follows:

$$\mathrm{FFN}\left(G'\right) = \mathrm{ReLU}\left(G'W_1 + b_1\right)W_2 + b_2. \tag{3}$$

Obtained by Equations (2) and (3), the output of the encoder denoted by $\widehat{G}$ is as follows:

$$\widehat{G} = \mathrm{Ins.Norm}\left(\mathrm{FFN}\left(G'\right) + G'\right). \tag{4}$$

By virtue of the self-attention structure, the global feature map of two consecutive frames can be aggregated to generate $\widehat{G}$. $\widehat{G}$ will be input as the common key into the next two shared decoders.

(2) *Decoder*

In the decoder, two decoders that share the network structure—detection decoder and track decoder, as shown in the orange connection and blue connection, are used to generate the player track boxes and the player detection boxes of the current frame, respectively. The orange line in Figure 2 represents the object decoder. The learnable detected player feature in the current frame is used as its input. We concatenate the representation of all the player's patches into an object query F_{object} in the current frame. As shown by the orange line in Figure 2, the first self-attention block, including residual connection and normalization, outputs the middle layer feature F'_{object}, which is expressed as follows:

$$F'_{\mathrm{object}} = \mathrm{Ins.Norm}\left(A_{F_{\mathrm{object}} \longrightarrow F_{\mathrm{object}}} F_{\mathrm{object}} + F_{\mathrm{object}}\right). \tag{5}$$

Based on the common feature $\widehat{G}$ in Equation (4) and the middle layer feature F'_{object} in Equation (5), we can compute the similarity matrix as follows:

$$A_{\widehat{G} \longrightarrow F'_{\mathrm{object}}} = \mathrm{Attention}\left(\widehat{G}, F'_{\mathrm{object}}\right). \tag{6}$$

Then, the cross-attention matrix $A_{\widehat{G} \longrightarrow F_{\mathrm{object}}}$ is fed to the residual connection and normalization layer as follows:

$$F'_{\mathrm{object}}{}' = \mathrm{Ins.Norm}\left(A_{\widehat{G} \longrightarrow F'_{\mathrm{object}}} \widehat{G} + F'_{\mathrm{object}}\right), \tag{7}$$

where $F'_{\mathrm{object}}{}'$ is the middle feature exported by the cross-attention block, including the residual connection and normalization. Furthermore, the feed forward network is added to the end of the decoder. In the detection encoder, we finally calculate the object feature, which is detected from the aggregated global feature map $\widehat{G}$ by the object query. This object feature $\widehat{F_{\mathrm{object}}}$ is the next frame's track query as follows:

$$\widehat{F_{\mathrm{object}}} = \mathrm{Ins.Norm}\left(\mathrm{FFN}\left(F'_{\mathrm{object}}{}'\right) + F'_{\mathrm{object}}{}'\right). \tag{8}$$

The blue connection represents the track decoder, as seen in Figure 2. The detected object feature map of the previous frame is fed to the track decoder as the track query F_{track} of the current frame. Similar to the first self-attention block in the aforementioned detection decoder, F'_{track} is the feature of the middle layer as follows:

$$F'_{\mathrm{track}} = \mathrm{Ins.Norm}\left(A_{F_{\mathrm{track}} \longrightarrow F_{\mathrm{track}}} F_{\mathrm{track}} + F_{\mathrm{track}}\right). \tag{9}$$

To leverage the temporal context information between the two consecutive frames and transform the temporal motion prior [27], we construct a Gaussian Radial Basis Function—temporal feature for the track query as follows:

$$m(y) = \exp\left(-\frac{\|y - c\|^2}{2\sigma^2}\right), \tag{10}$$

where c is the ground truth of the object position. Temporal mask matrix M_{temp} is the temporal feature ensemble. Similar to the aforementioned detection decoder, after the cross-attention block, we obtain the middle layer feature as follows:

$$F'_{\mathrm{track}}{}' = \mathrm{Ins.Norm}\left(A_{\widehat{G} \longrightarrow F_{\mathrm{track}}} \widehat{G} + F'_{\mathrm{track}} \otimes M_{\mathrm{temp}}\right), \tag{11}$$

where $\otimes$ is the elementwise multiplication. Finally, $F'_{\mathrm{track}}{}'$ is fed to the feed forward network block, and the final output of the track decoder is $\widehat{F_{\mathrm{track}}}$ as follows:

$$\widehat{F_{\mathrm{track}}} = \mathrm{Ins.Norm}\left(\mathrm{FFN}\left(F'_{\mathrm{track}}{}'\right) + F'_{\mathrm{track}}{}'\right). \tag{12}$$

3.2. Training. As shown in Figure 1, a pair of adjacent frames are fed to our model during training, which comes from the same clip video. If the input is a single picture, translation conversion and other operations must be performed on the original picture to generate an input that simulates adjacent frames. The output of the backbone is the feature map of the current frame. The encoder module takes the feature map of the previous frame and the feature map of the current frame as input. These features come from the output of the feature extraction module. The feature map of the previous frame is retained and reused for current tracking. Cross-attention is the information exchange between the encoder and the decoder. Both decoder operations are detection tasks. On the one hand, the loss of the detection branch is the same as that of DETR, and the set prediction loss is used for the distribution of the ground truth during training. Then, the regression and classification loss are calculated. On the other hand, for the branch of the track, there is no need to allocate ground truth again when there are track characteristics, and the classification and regression loss are directly calculated.

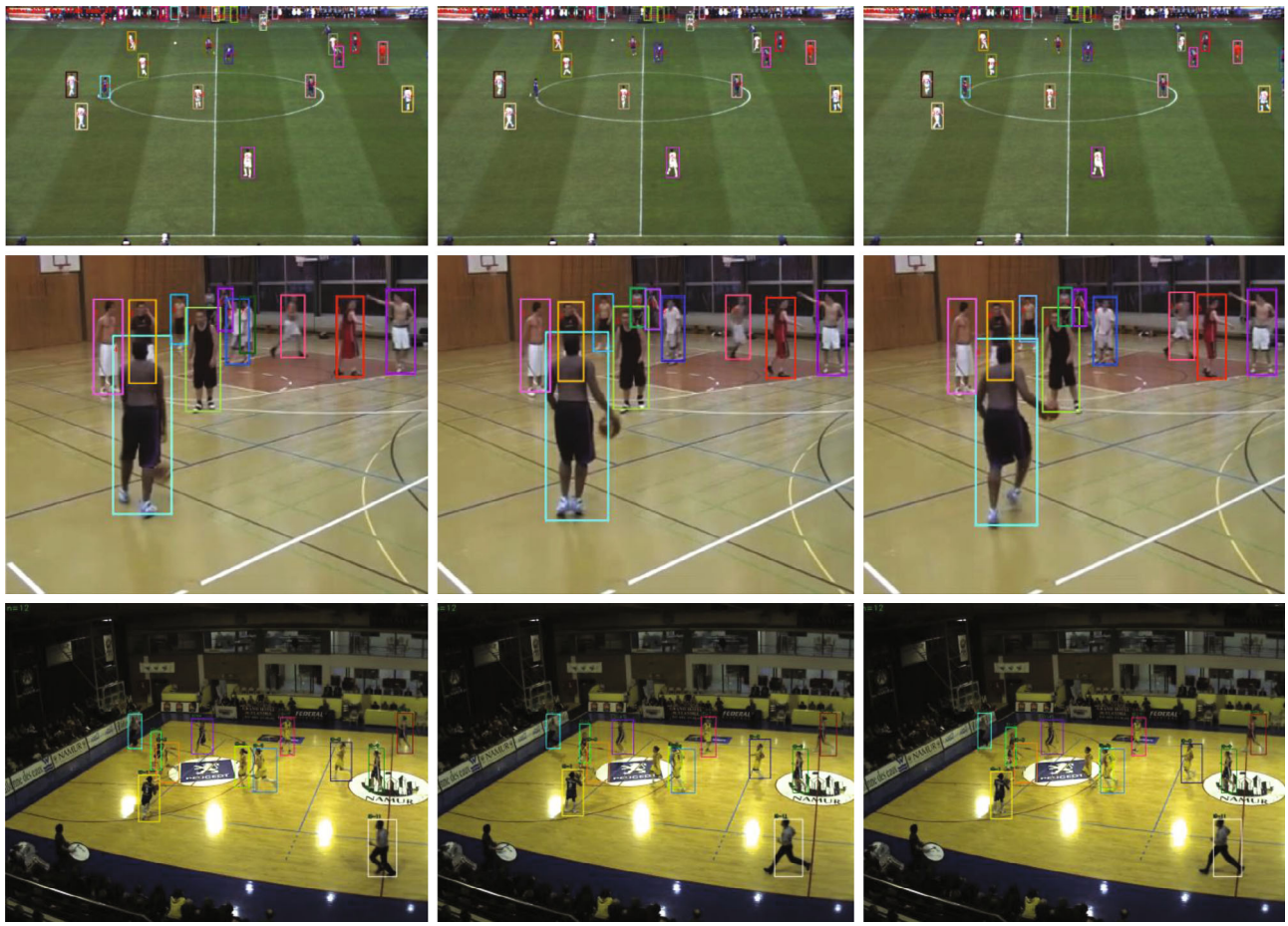

Figure 3: Sample tracking results from the APIDIS, EPFL-basketball, and ISSIA CNR-soccer datasets.

The final matching loss is defined as follows:

$$L_{\text{track}} = \lambda_{\text{cls}} \cdot L_{\text{cls}} + \lambda_{L_1} \cdot L_1 + \lambda_{\text{iou}} \cdot L_{\text{iou}}, \tag{13}$$

where L_{cls} is the focal loss of the predicted classifications and ground truth category labels. L_1 is the L_1 loss, and L_{iou} is the IoU loss between the normalized center coordinates and the height and width of the predicted boxes and the ground truth box. Then, λ_{cls}, λ_{L_1}, and λ_{iou} are the coefficients of each module.

During training, the output of the decoder will minimize the cross-entropy with the ground truth, but there is no correct answer in the inference process. To solve this problem and enable the model to detect new targets in time, we add exposure bias during the training process.

3.3. Inference. In the inference process, our method first obtains the global feature map of this frame through the CNN feature extractor and detects the players in it. Then, the two global feature maps of the first frame are combined and input into the model. After that, our method performs the object transform and box association for the next frames until the entire video sequence is completed. We use track rebirth tips [8], which are often used for tracking tasks to enhance the robustness where athletes' occlusions and movement directions change suddenly.

4. Experiments

To evaluate the proposed method, we present the experimental setting and performance compared with several MOT algorithms. There is no uniform evaluation standard for multiple tracking tasks. For fairness, we use the metrics in the MOT challenge. To verify that the encoder-decoder structure in our method makes good use of the information between the adjacent frames and the rebroadcasts of the tracking information of the previous frame to the current frame, we also perform ablation experiments.

4.1. Datasets

4.1.1. ISSIA-soccer. This dataset [28] is collected for a football match broadcast. The resolution is 1920*1080, and the frame rate is 25 fps. There are 6 videos, each of which is 2 minutes. At the beginning of every video, the first 300 frames without annotated information are used to remove the background initialization. Although the resolution of this dataset is high, the camera is far away from the target of tracking players, and the characteristics of the athletes are blurred.

4.1.2. APIDIS. The videos in this dataset [29] come from 7 fixed cameras, 5 of which are ordinary wide-angle cameras and 2 of which are fish-eye panoramic cameras taken from

Table 1: Comparison of tracking performance on the expanded APIDIS basketball dataset.

Methods	*Mode*	*MOTA*↑	*MOTP*↑	*IDF1*↑	*MT*↑	*ML*↓	*FP*↓	*FN*↓	*FPS*↓
CEM [36]	Offline	64.0	76.9	46.7	45.1	23.6	1598	3190	1.1
MHT [14]	Offline	73.2	78.6	50.3	54.7	22.9	890	2787	0.8
ELP [37]	Offline	74.7	80.4	55.9	55.3	21.4	801	2561	3.6
SORT [6]	Online	74.8	80.3	52.0	56.3	22.7	763	2757	17.2
Ours	Online	75.3	80.6	56.1	57.0	20.0	767	2538	15.1

Table 2: Comparison of tracking performance on the ISSIA-soccer dataset.

Methods	*Mode*	*MOTA*↑	*MOTP*↑	*IDF1*↑	*MT*↑	*ML*↓	*FP*↓	*FN*↓	*FPS*↓
CEM [36]	Offline	62.8	66.8	38.3	35.9	25.6	1062	2168	1.7
MHT [14]	Offline	63.5	68.2	39.8	36.2	22.1	306	2280	1.3
ELP [37]	Offline	67.5	65.1	43.6	38.9	24.3	334	2017	3.6
SORT [6]	Online	69.1	71.5	46.7	49.1	22.5	329	2033	17.2
Ours	Online	72.4	75.7	50.5	53.1	19.5	366	2480	23.2

Table 3: Comparison of tracking performance on the EPFL-basketball dataset.

Methods	*Mode*	*MOTA*↑	*MOTP*↑	*IDF1*↑	*MT*↑	*ML*↓	*FP*↓	*FN*↓	*FPS*↓
CEM [36]	Offline	59.0	63.3	37.9	48.0	41.1	2834	5863	5.4
MHT [14]	Offline	61.2	61.0	40.1	47.3	36.4	3037	5974	3.8
ELP [37]	Offline	66.4	65.9	45.3	45.6	39.0	2785	5359	6.0
SORT [6]	Online	67.1	72.9	52.9	53.8	38.8	2944	4990	21.5
Ours	Online	68.7	68.5	50.6	54.5	31.7	2767	4550	19.0

the top of the venue. The size is 1600×1200, and the frame rate is almost 22 fps on average. There are a total of 1500 frames in the dataset with tracking information, including 2 referees and 10 basketball players. However, this dataset has slightly fewer labeled frames. To enable the dataset to be trained in the deep network, we expand the annotation information. We choose a clip with a length of 30 s from 5 ordinary wide-angle camera original videos. Then, we supplement their tracking information. Finally, 4800 frames can be used for tracking training.

4.1.3. EPFL-basketball. The EPFL-basketball dataset [30] was taken in the school basketball hall, with 4 fixed cameras standing on the ground, shooting from the four directions of the basketball court. There are 4 segments, each of which is a 6-minute video taken from four views. The resolution is 360*288, and the frame rate is 25 fps. This dataset has a low resolution, which poses challenges for detection and tracking.

4.2. Implementation Details. In the feature extraction module, we use ResNet-34 [31] as the backbone. We use the parameters after pretrained on the COCO dataset [32]. For simplicity, we fix the weight of the pretrained ResNet-34 and only fine-tune the fully connect connected layers. We train our model with the Adam optimizer for 30 epochs with a starting learning rate of e^{-4}, and the learning rate decays to e^{-5} at 20 epochs. The batch size is set to 12. We use standard data augmentation techniques, including rotation, scaling, and color jittering.

4.3. Evaluation Metrics. There is no established standard of evaluation for multi-player tracking. For the sake of fairness, it is feasible to utilize the MOT metrics [33, 34] to measure the multiplayer tracking methods. Multiobject tracking accuracy (MOTA) mainly considers all object matching errors in tracking, including the ratio of misses in the sequence, false positives, and of mismatches. Multiobject tracking precision (MOTP) represents the accuracy of the target position. The closer MOTP is to 1, the higher the positioning accuracy of the tracker. Mostly tracked targets (MT) mean the ratio of groundtruth trajectories that are covered by a track hypothesis for at least 80% of their respective lifespan. Mostly lost targets (ML) represent the ratio of groundtruth trajectories that are covered by a track hypothesis for at most 20% of their respective lifespan [35]. False positive (FP) is the total number of false positive, and false negative (FN) expresses the total number of false negatives. The IDF1 score is the ratio of correctly identified detections to the average number of groundtruth and computed detections. Frame per second (FPS) indicates the speed of tracking processing. IDS is the number of ID switches, that is, the tracking object ID is different from its historical ID, which often occurs when multiple objects block each other.

Table 4: Ablation study on the temporal mask.

Temporal mask	*IDF1*↑	*MOTA*↑	*MOTP*↑	*MT*↑	*ML*↓
With	55.3	59.7	77.3	53.1	25.7
Without	56.1	75.3	80.6	57.0	20.0

Table 5: Temporal contexts between consecutive frames can improve the tracking effect on the expanded APIDIS basketball dataset.

Adding feature	*IDF1*↑	*MOTA*↑	*MOTP*↑	*MT*↑	*ML*↓
Current & Current	53.5	67.9	69.0	51.9	25.9
Translated & Current	50.1	66.7	74.1	55.8	23.4
Previous & Current	56.1	75.3	80.6	57.0	20.0

Table 6: Ablation experiment of normalization.

Normalization	*IDF1*↑	*MOTA*↑	*MOTP*↑
BatchNorm [38]	61.3	59.7	77.3
LayerNorm [39]	60.7	55.3	79.5
GroupNorm [40]	62.5	66.9	80.4
PowerNorm [41]	62.0	63.4	79.2
InstanceNorm [26]	56.1	75.3	80.6

4.4. Result Analysis. We first compare our method with other state-of-the-art multiple object tracking and multiple player tracking pipelines. An example of our multiplayer tracking result of the proposed method on three sports datasets is shown in Figure 3. Then, we verified the effect of the proposed method on multiplayer tracking for each component and evaluated our extracted temporal context tracking by comparing it to changing the input of the decoder.

4.4.1. Compared with Other Trackers. We compare the proposed method with other commonly classical multiobject tracking methods CEM [36], MHT [14], ELP [37], and SORT [6] on three datasets APIDIS, ISSIA-soccer, and EPFL-basketball. As shown in Tables 1–3, by virtue of the rich temporal context information between two consecutive frames, our method suppresses the comparative methods on MOTA and IDF1 on three datasets. Video is streaming media, and the temporal context information in it is crucial for continuous tracking.

However, most current trackers [14, 36, 37] tackle the task by frame-by-frame object tracking, where the temporal relationship between consecutive frames is largely ignored. Among them, SORT [6] records the object state from the previous frame in the object state management in the tracking stage, and this idea is widely adopted in the multiobject tracking model. Although historical frames are considered in some of the above methods, video frames are still considered independent and do not contribute to each other. We directly use the combined features of the previous frame and the current frame to generate tracking track boxes and use the interframe context to predict the position of the target of the previous frame in the current frame. The experimental results show that our method can better manage the problem of multiobject tracking in sports, such as for the players with similar appearances and complex movement states. However, our method does not perform as well as the previous research results on the FPS indicator because we introduce a more complex transformer structure in the pipeline, which increases its calculation time.

4.4.2. Temporal Mask. Due to the uniqueness of the attention mechanism, the transformer pays more attention to the drastically changing parts of the image, including the background that should not be highlighted. We leverage the temporal mask mechanism to suppress the background changes transformed from the previous frame temporally and to concentrate on the target player. As we can see from Table 4, the performance of our approach with the mechanism outperforms that without it, which proves that the effectiveness of the temporal mask mechanism can suppress the effect of changing the background on the tracking results.

4.4.3. Temporal Contexts. We also conduct comparative experiments on temporal contexts, as shown in Table 5.

Experimental results show that adding the object feature map of the previous frame has a better effect and reduces the probability of missing tracking. To verify that the transformer structure in our method is conducive to extracting the temporal contexts between consecutive frames, we test the use of different feature maps as the input of the encoder part, and the current feature map is combined into a composite. "Current & Current" means that the two current frame feature maps are used as a pair of input encoders. "Translated & Current" means that the current frame is subjected to random scaling and translating operations to the feature map and the original frame's feature map as input. "Previous & Current" is the combination of the global feature map of the previous frame and the current frame and the input encoder described above. As seen in Table 5, if the object feature map of the previous frame is not used, the MOTA result will be reduced by 7.4-8.6%.

4.4.4. Feature Normalization. If the value range of each dimension of the input matrix has a large difference, it will cause a large difference in the slope of the loss function in each direction, and training will become difficult. To address this dilemma, previous researchers have proposed a variety of effective feature normalization methods. To make the model training more efficient. We conduct comparative experiments on different normalization methods, and the results are shown in Table 6. Instance normalization is pixelwise to calculate the mean and the standard deviation, which corresponds to the pixel-to-pixel correspondence between the two frames in our method of cross-attention. The results show that instance normalization is more suitable for our player tracking task.

5. Conclusion

In this paper, we propose a novel model for multiplayer tracking in broadcast sports game videos. We take advantage of the transformer-based structure and make full use of the temporal contexts between consecutive frames. Extensive experiments are conducted to demonstrate that after adding the temporal context information, our model improves the results in the sports videos. Deep neural networks and transformer networks have achieved tremendous success in many vision applications. We will utilize the tracker for better sports video analysis.

Data Availability

We used the public datasets ISSIA-soccer, APIDIS, and EPFL-basketball, which are marked in the paper.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

The result of this paper is supported by the National Key R&D Program of China (Grant No. 2019YFB1406201).

References

[1] P. Nillius, J. Sullivan, and S. Carlsson, "Multi-target tracking - linking identities using Bayesian network inference," in *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, vol. 2, pp. 2187–2194, New York, NY, USA, 2006.

[2] J. Xing, H. Ai, L. Liu, and S. Lao, "Multiple player tracking in sports video: a dual-mode two-way Bayesian inference approach with progressive observation modeling," *IEEE Transactions on Image Processing*, vol. 20, no. 6, pp. 1652–1667, 2011.

[3] L. Kong, D. Huang, and Y. Wang, "Long-term action dependence-based hierarchical deep association for multi-athlete tracking in sports videos," *IEEE Transactions on Image Processing*, vol. 29, pp. 7957–7969, 2020.

[4] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: towards real-time object detection with region proposal networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 6, pp. 1137–1149, 2017.

[5] W. Kim, S.-W. Moon, J. Lee, D.-W. Nam, and C. Jung, "Multiple player tracking in soccer videos: an adaptive multiscale sampling approach," *Multimedia Systems*, vol. 24, no. 6, pp. 611–623, 2018.

[6] A. Bewley, Z. Ge, L. Ott, F. Ramos, and B. Upcroft, "Simple online and realtime tracking," in *2016 IEEE International Conference on Image Processing (ICIP)*, pp. 3464–3468, Phoenix, AZ, USA, 2016.

[7] N. Wojke, A. Bewley, and D. Paulus, "Simple online and real-time tracking with a deep association metric," in *2017 IEEE International Conference on Image Processing (ICIP)*, pp. 3645–3649, Beijing, China, 2017.

[8] P. Bergmann, T. Meinhardt, and L. Leal-Taixe, "Tracking without bells and whistles," in *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 941–951, Seoul, Korea (South), 2019.

[9] Y. Zhang, C. Wang, X. Wang, W. Zeng, and W. Liu, "FairMOT: on the fairness of detection and re-identification in multiple object tracking," *International Journal of Computer Vision*, vol. 129, no. 11, pp. 3069–3087, 2021.

[10] A. Vaswani, N. Shazeer, N. Parmar et al., "Attention is all you need," in *Proceedings of the 31st International Conference on Neural Information Processing Systems*, Long Beach, CA, USA, 2017.

[11] L. Chen, X. Peng, and M. Ren, "Recurrent metric networks and batch multiple hypothesis for multi-object tracking," *IEEE Access*, vol. 7, pp. 3093–3105, 2019.

[12] L. Leal-Taixé, C. Canton-Ferrer, and K. Schindler, "Learning by tracking: Siamese CNN for robust target association," in *2016 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pp. 418–425, Las Vegas, NV, USA, 2016.

[13] J. Son, M. Baek, M. Cho, and B. Han, "Multi-object tracking with quadruplet convolutional neural networks," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3786–3795, Honolulu, HI, USA, 2017.

[14] C. Kim, F. Li, A. Ciptadi, and J. M. Rehg, "Multiple hypothesis tracking revisited," in *2015 IEEE International Conference on Computer Vision (ICCV)*, pp. 4696–4704, Santiago, Chile, 2015.

[15] S. Anton Milan, H. Rezatofighi, A. Dick, I. Reid, and K. Schindler, "*Online multi-target tracking using recurrent neural networks*," in *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence (AAAI'17)*, pp. 4225–4232, San Francisco, CAL, USA, 2017.

[16] J. Yin, W. Wang, Q. Meng, R. Yang, and J. Shen, "A unified object motion and affinity model for online multi-object tracking," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 2020, pp. 6768–6777, Virtual, 2020.

[17] H. Zhou, W. Ouyang, J. Cheng, X. Wang, and H. Li, "Deep continuous conditional random fields with asymmetric inter-object constraints for online multi-object tracking," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 29, no. 4, pp. 1011–1022, 2019.

[18] V. Pallavi, J. Mukherjee, A. K. Majumdar, and Shamik Sural, "Graph-based multiplayer detection and tracking in broadcast soccer videos," *IEEE Transactions on Multimedia*, vol. 10, no. 5, pp. 794–805, 2008.

[19] M. Manafifard, H. Ebadi, and H. Abrishami Moghaddam, "A survey on player tracking in soccer videos," *Computer Vision and Image Understanding*, vol. 159, pp. 19–46, 2017.

[20] J. Liu, X. Tong, W. Li, T. Wang, Y. Zhang, and H. Wang, "Automatic player detection, labeling and tracking in broadcast soccer video," *Pattern Recognition Letters, Video-based Object and Event Analysis*, vol. 30, no. 2, pp. 103–113, 2009.

[21] J. Liu, P. Carr, R. T. Collins, and Y. Liu, "Tracking sports players with context-conditioned motion models," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1830–1837, Portland, OR, USA, 2013.

[22] A. Dosovitskiy, L. Beyer, A. Kolesnikov et al., "An image is worth 16x16 words: transformers for image recognition at scale," 2020, https://arxiv.org/abs/2010.11929.

[23] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "*BERT: pre-training of deep bidirectional transformers for language understanding*," in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 4171–4186, Minneapolis, Minnesota, 2019.

[24] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, "End-to-end object detection with transformers," in *16th European Conference on Computer Vision*, pp. 213–229, Cham, 2020.

[25] P. Sun, J. Cao, Y. Jiang et al., "TransTrack: multiple object tracking with transformer," 2020, https://arxiv.org/abs/2012.15460.

[26] D. Ulyanov, A. Vedaldi, and V. Lempitsky, "*Instance normalization: the missing ingredient for fast stylization*," 2017, https://arxiv.org/abs/ArXiv:1607.08022.

[27] N. Wang, W. Zhou, J. Wang, and H. Li, "Transformer meets tracker: exploiting temporal context for robust visual tracking," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1571–1580, Nashville, TN, USA, 2021.

[28] T. D'Orazio, M. Leo, N. Mosca, P. Spagnolo, and P. L. Mazzeo, "A semi-automatic system for ground truth generation of soccer video sequences," in *2009 Sixth IEEE International Conference on Advanced Video and Signal Based Surveillance*, pp. 559–564, Genova, Italy, 2009.

[29] C. De Vleeschouwer and D. Delannay, *Basket Ball Dataset from the European Project APIDIS*, 2009.

[30] F. Fleuret, J. Berclaz, R. Lengagne, and P. Fua, "Multicamera people tracking with a probabilistic occupancy map," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 30, no. 2, pp. 267–282, 2008.

[31] K. He, X. Zhang, S. Ren, and J. Sun, "*Deep residual learning for image recognition*," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, Las Vegas, NV, USA, 2016.

[32] T.-Y. Lin, M. Maire, S. Belongie et al., "Microsoft COCO: common objects in context," in *Computer Vision – ECCV 2014*, Lecture Notes in Computer Science, pp. 740–755, 2014.

[33] K. Bernardin and R. Stiefelhagen, "Evaluating multiple object tracking performance: the CLEAR MOT metrics," *EURASIP Journal on Image and Video Processing*, vol. 2008, Article ID 246309, 10 pages, 2008.

[34] E. Ristani, F. Solera, R. Zou, R. Cucchiara, and C. Tomasi, "Performance measures and a data set for multi-target, multi-camera tracking," in *Computer Vision – ECCV 2016 Workshops*, Lecture Notes in Computer Science, pp. 17–35, 2016.

[35] N. Ran, L. Kong, Y. Wang, and Q. Liu, "A robust multi-athlete tracking algorithm by exploiting discriminant features and long-term dependencies," in *MultiMedia Modeling*, Lecture Notes in Computer Science, I. Kompatsiaris, B. Huet, V. Mezaris, C. Gurrin, W.-H. Cheng, and S. Vrochidis, Eds., pp. 411–423, Springer International Publishing, 2019.

[36] A. Milan, S. Roth, and K. Schindler, "Continuous energy minimization for multitarget tracking," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 36, no. 1, pp. 58–72, 2014.

[37] N. McLaughlin, J. M. Del Rincon, and P. Miller, "Enhancing linear programming with motion modeling for multi-target tracking," in *2015 IEEE Winter Conference on Applications of Computer Vision*, pp. 71–77, Santiago, Chile, 2015.

[38] S. Ioffe and C. Szegedy, "Batch normalization: accelerating deep network training by reducing internal covariate shift," in *Proceedings of the 32nd International Conference on Machine Learning*, pp. 448–456, Lille, France, 2015.

[39] J. L. Ba, J. R. Kiros, and G. E. Hinton, "Layer normalization," 2016, https://arxiv.org/abs/1607.06450.

[40] Y. Wu and K. He, "Group normalization," in *Proceedings of the European conference on computer vision (ECCV)*, pp. 3–19, Munich, Germany, 2018.

[41] S. Shen, Z. Yao, A. Gholami, M. W. Mahoney, and K. Keutzer, "PowerNorm: rethinking batch normalization in transformers," in *Thirty-seventh International Conference on Machine Learning*, Vienna, Austria, 2020.